CES-11
Prof. Armando

5º Exercício:
Agenda Eletrônica + Heap

Pessoas ativas costumam ter dificuldade na gestão de tempo, para conciliar o trabalho, os estudos e outras atividades como iniciativas e eventos. ☺

Para auxiliar nesse quesito, vamos elaborar um programa que implementa um TAD **fila de prioridade** para ajudar na organização de tarefas de uma agenda eletrônica.

O usuário insere as suas atividades na agenda e atribui a cada uma delas um valor de importância, no mínimo 1 e no máximo 99. As tarefas podem ser inseridas a qualquer momento do dia. Quando o usuário termina uma tarefa, clica na agenda para consultar qual a tarefa mais importante a ser realizada em seguida.

Ao terminar o expediente, a agenda permite que o usuário selecione o comando "FIM". Nessa ocasião, o programa deve desfazer a fila e imprimir, para consulta, todas as tarefas remanescentes, ordenadas por importância.

Apresente na saída:

Mensagens de aviso:

- se o usuário consultar a próxima tarefa e a agenda estiver vazia;
- se no final a agenda estiver vazia;

Mensagem de erro:

- se houver tentativa de inserir mais elementos que a capacidade da agenda.

As mensagens são livres. Mas devem estar alinhadas conforme o exemplo.

Arquivos:

Entrada: `entrada5.txt`
Saída: `Lab5_seu_nome_e_sobrenomes.txt`
Os dois arquivos na mesma pasta que o programa.

Observações:

O cabeçalho do arquivo de entrada será conforme o exemplo.
O cabeçalho do arquivo de saída será conforme o exemplo.
Tamanho de cada linha nos cabeçalhos: máximo 70 caracteres.
Tamanho da descrição: máximo 40 caracteres.
Tarefas diferentes podem ter mesmo valor de prioridade, sim.
Nesse caso, a agenda pode escolher qualquer uma delas.
A saída deve seguir o **alinhamento** no formato do exemplo.

A programação do TAD fila de prioridade deve ser da seguinte forma.

Encapsulamento

Neste programa você deve manter "escondida" a estrutura de dados,
deve fazer implementação de diversas funções referentes aos operadores
e deve seguir restrições de acesso ao TAD.

Restrições de acesso

Não usar variáveis globais.
Apenas a função main pode ler e/ou escrever nos arquivos de entrada e saída.
A função main não pode acessar diretamente os campos dentro do TAD.
A função main pode acessar o TAD somente através das funções "públicas".

Funções públicas

O programa deve conter e utilizar funções separadas para cada operação no TAD:

Inserir
ConsultarMax
RemoverMax (deve ser `void`)
Inicializar
Finalizar (deve somente dar `free`)
FilaVazia (retorna `bool`)
FilaCheia (retorna `bool`)

Funções privadas

Se quiser, pode fazer funções extras, que podem acessar o TAD.
Mas a função main não pode usar as funções privadas.

Estruturas de dados: heap

A lista com os elementos (tarefas) deve ficar armazenada em um
heap implementado com vetor, onde:

a posição 0 não é utilizada;
a posição 1 contém a raiz do heap.

A variável do heap deve ser uma struct contendo:

o vetor;
quantidade de elementos;
quantidade máxima possível de elementos (será lida do arquivo de entrada).

Encapsulamento - parte 2

Neste programa você deve obrigatoriamente utilizar
as declarações de tipos, a função main e os cabeçalhos de funções
fornecidos no arquivo anexo.